Tutorial on Boosted Final States

CTEQ School 2013

July 15, 2013

1 Introduction

In this tutorial we will discuss boosted final states, using the example of a Standard Model Higgs boson decaying to two b-jets. You may work in groups, where each member generates predictions from a different event generator. Ideally, you will have a selection of people who already know "their" generator from using it in the first tutorial. For instructions on how to exchange results, please review yesterday's tutorial worksheet.

2 General considerations

In this tutorial we are going to look at a boosted Higgs boson analysis. We will follow the analysis presented in [1], which aims at isolating the Higgs boson signal in the reaction $pp \to VH$, with the Higgs decaying to a pair of b-jets. The fact that the Higgs is boosted is essential in reducing backgrounds. The analyses are adapted from those used in [2].

Background samples are difficult to generate within the time frame of the tutorial, and they have therefore been prepared for you. In this exercise we will run the generators only for the signal. We will perform the study at the parton level, but after running the shower. A complete hadron-levele analysis is left to you as an exercise.

Following [1], we will consider three selection criteria which identify high transverse momentum vector bosons.

Selection A An e^+e^- pair or $\mu^+\mu^-$ pair with an invariant mass between 80 GeV and 100 GeV with a transverse momentum larger $p_T^{\min} = 200$ GeV. Further leptons are vetoed.

Selection B Missing transverse momentum larger than $p_T^{\min} = 200$ GeV

Selection C Missing transverse momentum 30 GeV plus a lepton (e or μ) with transverse momentum > 30 GeV consistent with a W boson of nominal mass with $p_T^W > p_T^{\min} = 200$ GeV.

Only particles with rapidity between -2.5 and 2.5 are considered for constructing jets and leptons.

To suppress the top background we veto on additional b-jets. We will assume a b-tagging efficiency of 100% and a uniform mistag probability of 2%. Introducing a more realistic b-tagging efficiency of 60% and studying its influence on the significance of the signal is left to you as an additional exercise. To allow the identification of b-jets we prevent the decay of b-flavored hadrons in the hadron-level setups of the generators (see below).

3 Generating and analyzing events

First you need to build the Rivet analysis plugins. Change to the directory `~/tutorials/boost/analysis/` and run

```
make
```

This will build a Rivet plugin library from the source files in this directory. You will modify the sources at a later time.

3.1 Background samples

The backgrounds we are going to consider are $pp \to V + \mathrm{jets}$, $pp \to VV + \mathrm{jets}$ and $pp \to t\bar{t} + \mathrm{jets}$, where V can stand for either $W^{\pm}$ or Z.

An event file has been prepared for you, which is stored in `~/tutorials/boost/bg/`. Change to this directory. In order not to expand the large file, we use a fifo pipe, which is created as

```
mkfifo events
```

Now you can unzip the event file and, at the same time, run rivet to analyze the events:

```
bzcat bkg.500k.hepmc.bz2 > events & \
rivet -a HIGGSANALYSISA_JS -a HIGGSANALYSISB_JS -a HIGGSANALYSISC_JS -H bg.aida events
```

The analysis of these events should take about fifteen minutes. You can shorten it by analyzing fewer events (consider the help options of Rivet for a way to do that), but your final plots may have insufficient statistics in this case.

3.2 Herwig

Generate signal event samples or `.aida` files with HERWIG++. The directory `boost/herwig` contains a sample signal input file `LO_VH.in`. You can use LO or NLO matrix elements, switch the shower and hadronization on or off and also look at the effect of the underlying event. Take the input files from day 2 as a starting point. You should have discovered all relevant options in the tutorial sessions of the first two days.

3.3 Pythia

In this session, we will study the influence of the signal modelling on a boosted analysis. The materials for this tutorial can be found in

```
BOOSTDIR=~/tutorials/boost/pythia .
```

This directory contains the LHE files

`hwp_tree_0.lhe.gz`:	$pp \to HW^+$ (LO, $\mu_f = m_H, \mu_r = m_H + m_W$)
`hwp_tree_1.lhe.gz`:	$pp \to HW^+ + 1$ parton (LO, $\mu_f = m_H, \mu_r = m_H + m_W$)
`hwp_tree_2.lhe.gz`:	$pp \to HW^+ + 2$ partons (LO, $\mu_f = m_H, \mu_r = m_H + m_W$) ,

the settings files

`main85.cmnd`:	Input file for CKKW-L merging
`main-no-merging.cmnd`:	Input file for default PYTHIA 8 (i.e. simply showering $pp \to HW^+$),

the sample main programs

`main85.cc`:	Main program for CKKW-L merging.
`main-no-merging.cc`:	Main program for default PYTHIA 8 (i.e. simply showering $pp \to HW^+$),

and the script `run.sh`. To start this session, change into `BOOSTDIR` and compile the main programs:

```
cd $BOOSTDIR
make main85
make main-no-merging
```

Now have a look at `run.sh`. This is very similar to the script we have used for the Higgs / NLO merging session: It runs PYTHIA 8, and directly feeds the output to RIVET by using a `fifo` pipe. Use this script to

- Histogram the output of showering $pp \to HW^+$.

- Histogram the output of showering $pp \to HW^+$, where $W^+ \to e^+\nu_e, \mu^+\nu_\mu$. For this, you will have to add the settings
  ```
  24:onMode = off
  24:onIfAny = 11 13
  ```
  where the first command switches off all W-boson decays, and the second enables only the decay modes containing electrons or muons.

- Histogram the output of CKKW-L merging with up to two additional jets, with $W^+ \to e^+\nu_e, \mu^+\nu_\mu$.

What differences do you find? For which settings is the Higgs mass peak most significant, and which simulations profit most from the boosted analysis? You can also try to change further settings (e.g. switch off multi-parton interactions). Finally, plot the results of the signal simulation together with the background.

Once you have understood the results, and if you still have time left, try changing the analysis to improve the significance of the mass peak (see e.g. section 4). Bear in mind that, with some simulation setups, you might overestimate the gain.

3.4 Sherpa

Change to the tutorial directory `tutorials/boost/sherpa`. Run Sherpa to generate process-specific information and compute the cross sections:

```
Sherpa
```

While Sherpa runs, have a look at its input file. In the `(processes)` section you will find all entries needed to simulate the signal in a pure parton shower approach, or with LO merging methods using up to `NJETS` additional jets. Note that we produce stable electroweak bosons, which are decayed using Sherpa's internal decay package (cf. the `(run)` section).

The `(run)` section contains some new settings.

- We enable spin correlations in W boson decays using the parameter `HARD_SPIN_CORRELATIONS 1;`

- At the hadron level (see additional exercises) we disable B-hadron decays using `STABLE[5..] 1;`

Now it is your turn! Generate the following event files:

- A parton-shower prediction, defined by the settings in the runcard.

- A LO merged prediction with up to one additional jet.

Each run will take about ten minutes. Note that, because of the setting `EVENT_OUTPUT HepMC_Short[Signal_NJETj];` Sherpa will produce HepMC event files, which are labeled by the number of additional jets.

Analyze these event files by extracting them to a fifo pipe and running Rivet simultaneously. For example, in the zero jet case:

```
mkfifo events
gunzip -c Signal_0j.hepmc.gz > events & \
rivet -a HIGGSANALYSISA_JS -a HIGGSANALYSISB_JS -a HIGGSANALYSISC_JS -H sherpa_0j.aida events
```

Use the `plotit.sh` script or run `rivet-mkhtml` to plot your results. Which differences do you find between the merged sample and the parton-shower result?

You can now modify the input file to include hadronization and underlying event simulation. You can also try to improve the significance of the analysis, see Sec. 4.

4 Additional tasks

If you have time left during the tutorial, or as an additional exercise, you may want to do the following studies.

4.1 Trimming, Pruning and Filtering

Fastjet implements different substructure tools. The first analysis you used employed filtering, which keeps only a few subjets. You can use other tools as follows:

Try trimming with

```
fastjet::Selector sel_aboveptfrac = fastjet::SelectorPtFractionMin(0.05);
fastjet::Filter trimmer(Rfilt,sel_aboveptfrac);
```

Try pruning with

```
double zcut = 0.1;
double rcut_factor = 0.5;
fastjet::Pruner pruner(fastjet::cambridge_algorithm, zcut, rcut_factor);
filteredJet=pruner(tagged_jet);
```

You need to rebuild the rivet analyses after this modification, see Sec. 3. Please remember to write the output of the new analysis to a different file using the -H option of Rivet. You can then compare the effect of different tools using rivet-mkhtml.

References

[1] J. M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, Phys. Rev. Lett. **100** (2008) 242001 [arXiv:0802.2470 [hep-ph]].

[2] P. Richardson and D. Winn, Eur. Phys. J. C **72** (2012) 2178 [arXiv:1207.0380 [hep-ph]].